

**AMENDMENTS TO THE CLAIMS**

Upon entry of this amendment, the following listing of claims will replace all prior versions and listings of claims in the pending application.

Please amend claims 1-13, 16-21 and 22-32 as follows:

1. (Currently Amended) A method for virtualizing access to named system objects, the method comprising instructing a suitably programmed computer to perform the steps of:

- (a) receiving a request to access a system object stored in a memory element provided by a computer, the request received from a process executing in the context of a user isolation scope an isolation environment, the isolation environment comprising an application isolation layer and a user isolation layer, the request including a virtual name for the system object;
- (b) determining selecting, from a memory element provided by the computer, a rule associated with the request, the selection responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes;
- (c) forming a literal name for the system object in response to the determined rule; and
- (d) issuing to the operating system a request to access the system object, the request including the literal name for the system object.

2. (Currently Amended) The method of claim 1 wherein step (a) comprises receiving a request to access a system object stored in the memory element provided by the computer, the request received from a process executing in the context of a user isolation scope an isolation environment, the isolation environment comprising an application isolation layer and a user isolation layer, to access a system the object selected from the group consisting of a semaphore, a mutex, a mutant, a timer, an event, a job object, a file-mapping object, a section, a named pipe, and a ~~mailslot~~ mailslot, the request including a virtual name for the system object.

3. (Currently Amended) The method of claim 1 wherein step (a) comprises intercepting a request to access a system object from a process executing in the context of ~~a user isolation scope~~ an

isolation environment, the isolation environment comprising an application isolation layer and a user isolation layer, the request including a virtual name for the system object.

4. (Currently Amended) The method of claim 1 wherein step (a) comprises receiving a request from a process executing in the context of ~~a user isolation scope~~ an isolation environment, the isolation environment comprising an application isolation layer and a user isolation layer, to open a system object, the request including a virtual name for the system object.

5. (Currently Amended) The method of claim 1 wherein step (a) comprises receiving a request from a process executing in the context of ~~a user isolation scope~~ an isolation environment, the isolation environment comprising an application isolation layer and a user isolation layer, to create a system object, the request including a virtual name for the system object.

6. (Currently Amended) The method of claim 1 wherein step (b) comprises determining, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes, that a rule action selected from the group consisting of ignore, redirect and isolate, is associated with the request.

7. (Currently Amended) The method of claim 1 wherein step (b) comprises accessing a rules engine to determine, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes, a rule action associated with the virtual name included in the received request.

8. (Currently Amended) The method of claim 1 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes, a literal name for the system object stored in the memory element provided by the computer using the virtual name provided in the request and a scope-specific identifier.

9. (Currently Amended) The method of claim 8 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in

which the process executes, a literal name for the system object stored in the memory element provided by the computer using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with an application isolation scope with which the process making the request is associated.

10. (Currently Amended) The method of claim 8 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes, a literal name for the system object stored in the memory element provided by the computer using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with the user isolation scope in which the process making the request executes.

11. (Currently Amended) The method of claim 1 wherein step (c) further comprises the step of forming a literal name for the system object stored in the memory element provided by the computer identifying the system object as having global visibility.

12. (Currently Amended) The method of claim 1 wherein step (c) further comprises the step of forming a literal name for the system object stored in the memory element provided by the computer identifying the system object as having session visibility.

13. (Currently Amended) The method of claim 1 wherein step (c) comprises forming a literal name for the system object stored in the memory element provided by the computer that is ~~substantially~~ identical to the virtual name provided in the request.

14. (Original) The method of claim 1 further comprising the step of receiving a handle from the operating system identifying the accessed object.

15. (Original) The method of claim 14 further comprising the step of transmitting the handle to the process.

16. (Currently Amended) The method of claim 1 further comprising the step of receiving a request to access the system object from a second process executing in the context of a second isolation environment comprising an application isolation layer and a user isolation scope layer, the request including the virtual name for the object.

17. (Currently Amended) The method of claim 16 wherein step (c) comprises forming, responsive to the application isolation layer and the second user isolation layer forming an isolation environment in which the second process executes, a literal name for the system object using the virtual name provided in the request and a scope-specific identifier.

18. (Currently Amended) The method of claim 17 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes, a literal name for the system object stored in the memory element provided by the computer using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with an application isolation scope with which the process making the request is associated.

19. (Currently Amended) The method of claim 17 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes, a literal name for the system object stored in the memory element provided by the computer using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with the second user isolation scope in which the process making the request executes.

20. (Currently Amended) The method of claim 16 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes, a literal name for the system object stored in the memory element provided by the computer that is ~~substantially~~ identical to the virtual name provided in the request.

21. (Currently Amended) The method of claim 1 further comprising the step of receiving a request to access the system object from a second process executing in the context of the user isolation scope layer, the request including the virtual name for the object.

22. (Currently Amended) The method of claim 21 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the second process executes, a literal name for the system object using the virtual name provided in the request and a scope-specific identifier.

23. (Currently Amended) The method of claim 22 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the second process executes, a literal name for the system object using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with an application isolation scope with which the second process making the request is associated.

24. (Currently Amended) The method of claim 22 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the second process executes, a literal name for the system object using the virtual name provided in the request and a scope-specific identifier, the scope-specific identifier associated with the user isolation scope in which the second process making the request executes.

25. (Currently Amended) The method of claim 21 wherein step (c) comprises forming, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the second process executes, a literal name for the system object that is substantially identical to the virtual name provided in the request.

26. (Currently Amended) An apparatus for virtualizing access to named system objects comprising:

~~computer-readable program means for a hooking mechanism~~ receiving a request to access a system object from a process executing in the context of ~~a user isolation scope~~ an isolation environment, the isolation environment comprising an application isolation layer and a user isolation layer, the request including a virtual name for the system object;  
~~computer-readable program means for a name virtualization engine~~ forming a literal name for the system object responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes; and  
~~computer-readable program means for an operating system interface~~ requesting access to the system object using the literal name.

27. (Currently Amended) The apparatus of claim 26 wherein ~~the hooking mechanism~~ the computer-readable program means for receiving a request intercepts a request to open a system object.

28. (Currently Amended) The apparatus of claim 26 wherein ~~the hooking mechanism~~ the computer-readable program means for receiving a request intercepts a request to create a system object

29. (Currently Amended) The apparatus of claim 26 further comprising ~~a rules engine~~ computer-readable program means for storing a rule associated with the request.

30. (Currently Amended) The apparatus of claim 29 wherein the ~~rules engine~~ computer-readable program means for storing a rule comprises a database.

31. (Currently Amended) The apparatus of claim 26 wherein the ~~name virtualization engine~~ computer-readable program means for forming a literal name for the system object forms, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes, a literal name for the system object that is substantially identical to the virtual name.

32. (Currently Amended) The apparatus of claim 26 wherein the ~~name-virtualization-engine~~ computer-readable program means for forming a literal name for the system object forms, responsive to the application isolation layer and the user isolation layer forming the isolation environment in which the process executes, a literal name for the system object using the virtual name and a scope-specific identifier.

33. (Original) The apparatus of claim 32 wherein the scope-specific identifier is associated with an application isolation scope with which the process making the request is associated.

34. (Original) The method of claim 32 wherein the scope-specific identifier is associated with the user isolation scope in which the process making the request executes.